

//Q1. Write a C function to calculate the power of a number x^r .

```
#include <stdio.h>
```

```
int getPower(int x, int n) // function to get x to the power r
```

```
{  
    int pow = 1, i;  
    for(i=1; i<=n ; i++){  
        pow = pow * x;  
    }  
}
```

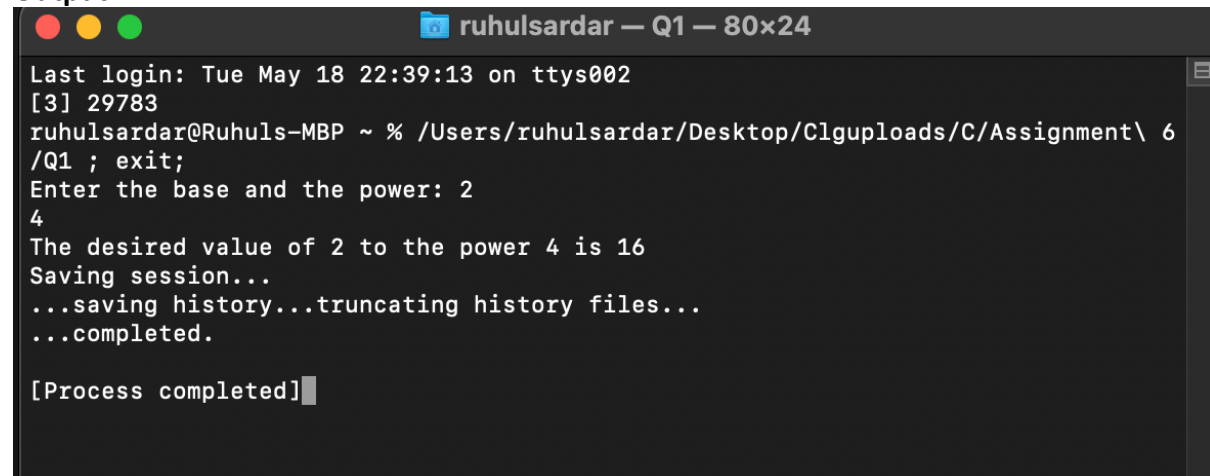
```
    return pow;  
}
```

```
int main(){
```

```
    int x, r , res;  
    printf("Enter the base and the power: ");  
    scanf("%d%d", &x, &r);  
    res = getPower(x,r);  
    printf("The desired value of %d to the power %d is %d", x,r,res);
```

```
    return 0;  
}
```

Output:



```
ruhulsardar — Q1 — 80x24  
Last login: Tue May 18 22:39:13 on ttys002  
[3] 29783  
ruhulsardar@Ruhuls-MBP ~ % /Users/ruhulsardar/Desktop/Clguploads/C/Assignment\ 6  
/Q1 ; exit;  
Enter the base and the power: 2  
4  
The desired value of 2 to the power 4 is 16  
Saving session...  
...saving history...truncating history files...  
...completed.  
[Process completed]
```

//Q2. Write a C function to calculate the binary equivalent of a decimal number.

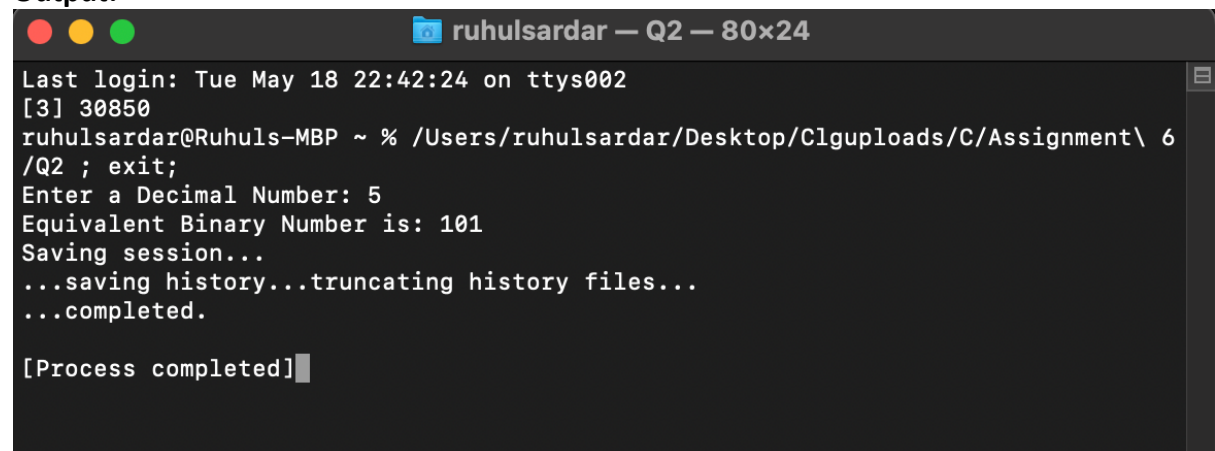
```
#include <stdio.h>
#include <math.h>

long decimalToBinary(int decimalnum)//function to get the binary Equivalent of the decimal
number.
{
    long binarynum = 0;
    int rem, temp = 1;

    while (decimalnum!=0)
    {
        rem = decimalnum%2;
        decimalnum = decimalnum / 2;
        binarynum = binarynum + rem*temp;
        temp = temp * 10;
    }
    return binarynum;
}

int main()
{
    int decimalnum;
    printf("Enter a Decimal Number: ");
    scanf("%d", &decimalnum);
    printf("Equivalent Binary Number is: %ld", decimalToBinary(decimalnum));
    return 0;
}
```

Output:



```
ruhulsardar — Q2 — 80x24
Last login: Tue May 18 22:42:24 on ttys002
[3] 30850
ruhulsardar@Ruhuls-MBP ~ % /Users/ruhulsardar/Desktop/Clguploads/C/Assignment\ 6
/Q2 ; exit;
Enter a Decimal Number: 5
Equivalent Binary Number is: 101
Saving session...
...saving history...truncating history files...
...completed.

[Process completed]
```

//Q3. Write a complete C program to find the greatest common divisor (gcd) of two positive integers are received as arguments to the function.

```
#include <stdio.h>
int gcd(int x, int y);

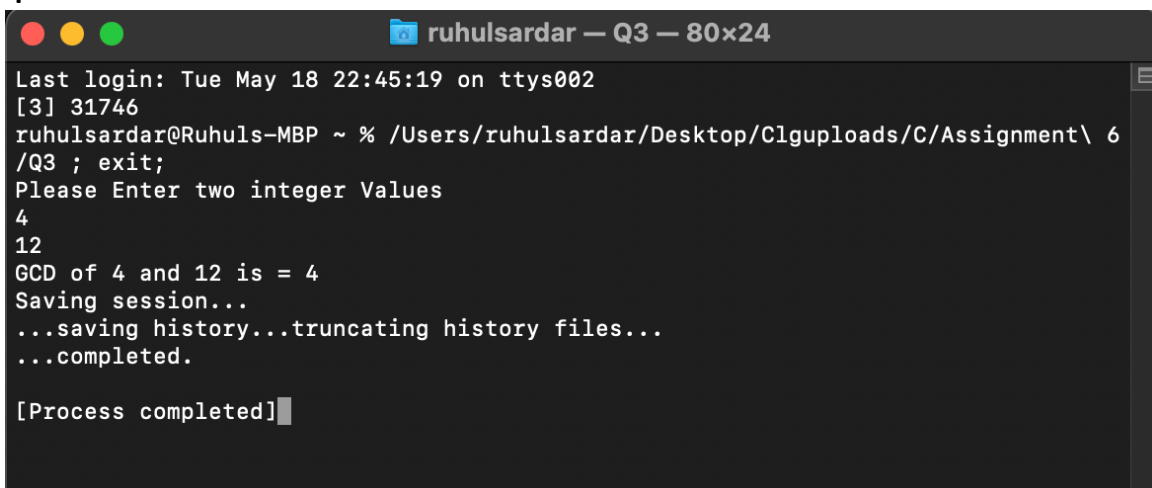
int main()
{
    int Num1, Num2;

    printf("Please Enter two integer Values \n");
    scanf("%d %d", &Num1, &Num2);

    printf("GCD of %d and %d is = %d", Num1, Num2, gcd(Num1, Num2));
    return 0;
}

int gcd(int x, int y)//function to calculate the gcd of 2 numbers.
{
    if (x == 0) {
        return y;
    }
    while (y != 0) {
        if (x > y) {
            x = x - y;
        }
        else {
            y = y - x;
        }
    }
    return x;
}
```

Output:



```
ruhulsardar - Q3 - 80x24
Last login: Tue May 18 22:45:19 on ttys002
[3] 31746
ruhulsardar@Ruhuls-MBP ~ % /Users/ruhulsardar/Desktop/C/guploads/C/Assignment\ 6
/Q3 ; exit;
Please Enter two integer Values
4
12
GCD of 4 and 12 is = 4
Saving session...
...saving history...truncating history files...
...completed.

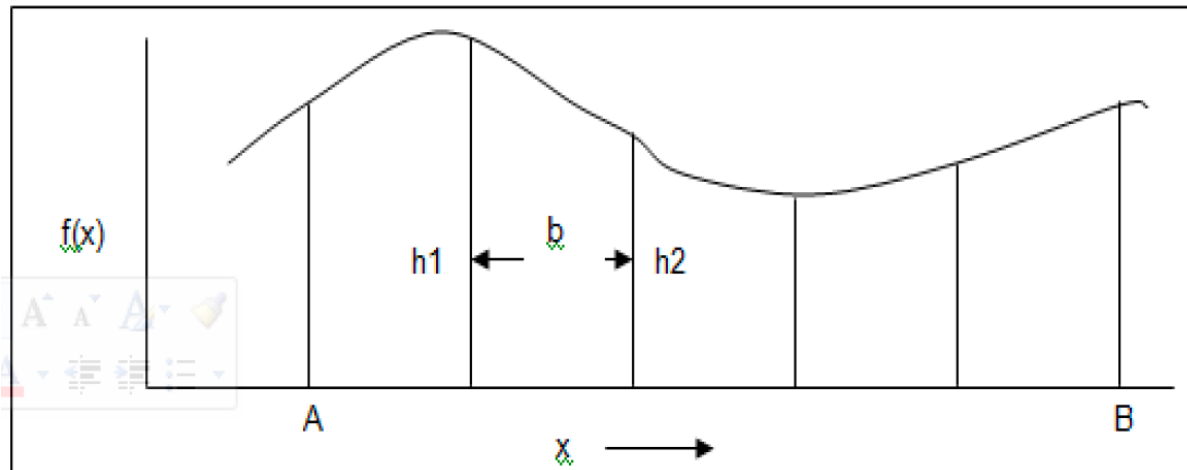
[Process completed]
```

//Q4.

One of the applications of computers in numerical analysis is computing the area under a curve. One simple method of calculating the area under a curve is to divide the area into a number of trapezoids of same width and summing up the area of individual trapezoids. The area of a trapezoid is given by –

$$\text{Area} = 0.5 \cdot (h_1 + h_2) \cdot b$$

where h_1 and h_2 are the heights of two sides and b is the width as shown in fig. below.



WACP to calculate the area for a curve of the function $f(x) = x^2 + 1$ between any two given say A and B as shown in figure above.

Hint: Inputs to the program are lower limit(A), upper limit(B) and the number of trapezoids

```
#include<stdio.h>
```

```
#include<math.h>
```

```
float f(float x)//function get the  $f(x) = x^2 + 1$ .
```

```
{  
    return(1+pow(x,2));  
}
```

```
int main()
```

```
{  
    int i,n;  
    float x0,xn,h,y[20],so,se,ans,x[20];  
    printf("\n Enter values of x0,xn,h:\n");  
    scanf("%f%f%f",&x0,&xn,&h);  
    n=(xn-x0)/h;  
    if(n%2==1)  
    {  
        n=n+1;  
    }  
}
```

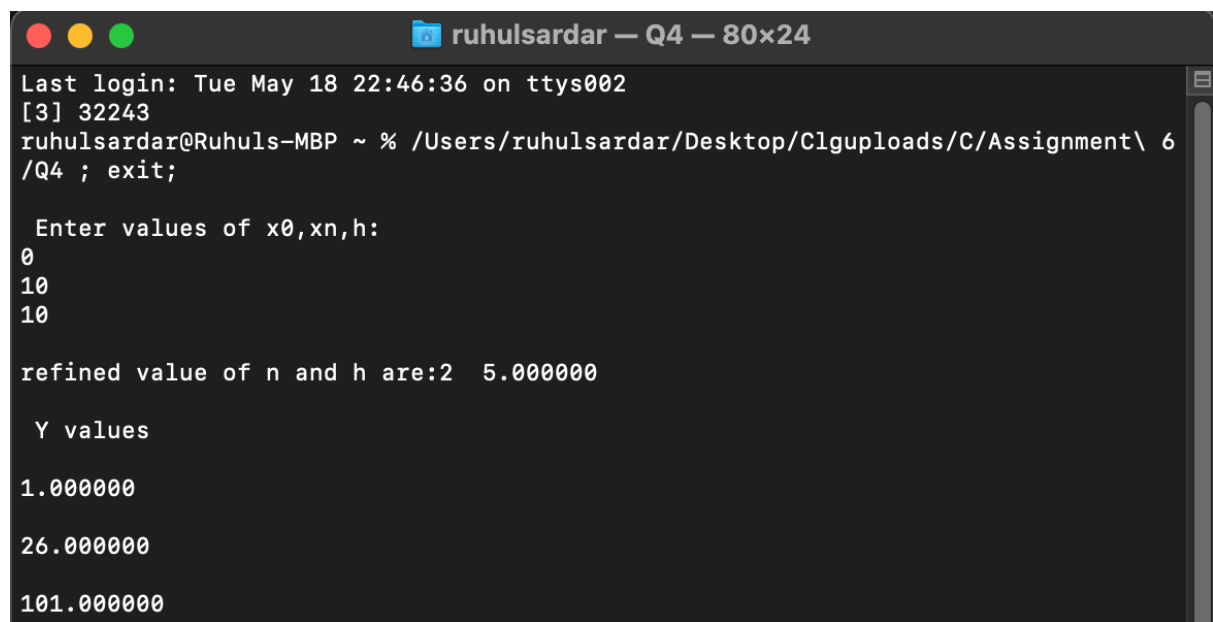
```

h=(xn-x0)/n;
printf("\nrefined value of n and h are:%d %f\n",n,h);
printf("\n Y values \n");
for(i=0; i<=n; i++)
{
    x[i]=x0+i*h;
    y[i]=f(x[i]);
    printf("\n%f\n",y[i]);
}
so=0;
se=0;
for(i=1; i<n; i++)
{
    if(i%2==1)
    {
        so=so+y[i];
    }
    else
    {
        se=se+y[i];
    }
}
ans=h/3*(y[0]+y[n]+4*so+2*se);
printf("\nfinal integration is %f",ans);

return 0;
}

```

Output:



A terminal window titled "ruhulsardar — Q4 — 80x24" displays the following output:

```

Last login: Tue May 18 22:46:36 on ttys002
[3] 32243
ruhulsardar@Ruhuls-MBP ~ % /Users/ruhulsardar/Desktop/Clguploads/C/Assignment\ 6
/Q4 ; exit;

Enter values of x0,xn,h:
0
10
10

refined value of n and h are:2  5.000000

Y values

1.000000

26.000000

101.000000

```

//Q5. . Write a C program to find a given number is pronic number or not.

```
#include <stdio.h>
#include <math.h>
#include <stdbool.h>

bool checkPronic(int x)// function to check wheather the given number is pronic or not.
{
    for (int i = 0;
        i <= (int)(sqrt(x));
        i++)

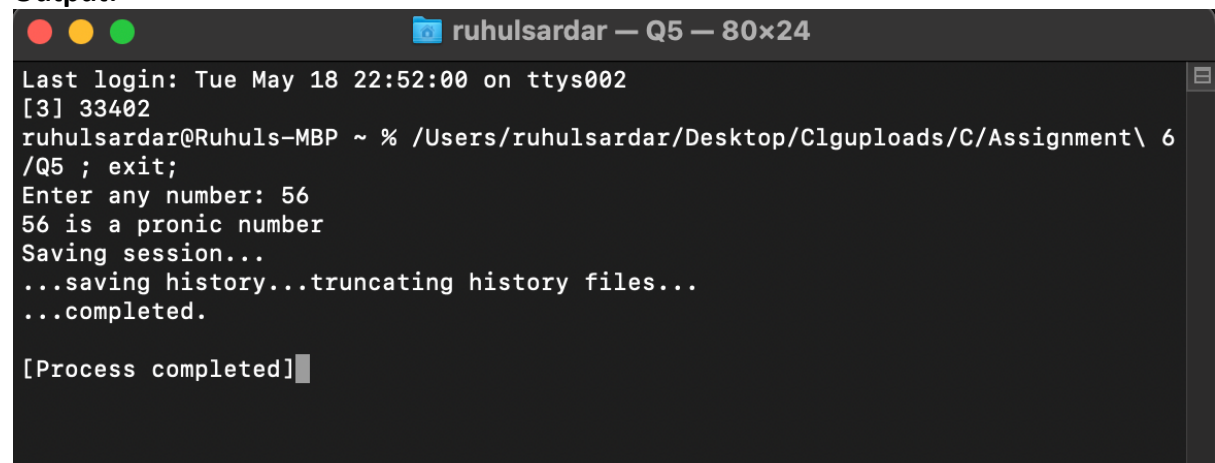
        if (x == i * (i + 1))
            return true;

    return false;
}

int main()
{
    int n, res;
    printf("Enter any number: ");
    scanf("%d", &n);
    res = checkPronic(n);
    if(res == 1)
        printf("%d is a pronic number", n);
    else
        printf("%d is not a pronic number", n);

    return 0;
}
```

Output:



```
ruhulsardar — Q5 — 80x24
Last login: Tue May 18 22:52:00 on ttys002
[3] 33402
ruhulsardar@Ruhuls-MBP ~ % /Users/ruhulsardar/Desktop/Clguploads/C/Assignment\ 6
/Q5 ; exit;
Enter any number: 56
56 is a pronic number
Saving session...
...saving history...truncating history files...
...completed.

[Process completed]
```